Qn1

a) The following are a couple programming languages that I like and why I like them:

i) JavaScript: JavaScript can also be run outside of the browser with the use of a framework like Node.js, Nashorn, Wakanda, or Google Apps Script. JavaScript is a fast and easy to use to accomplish a task. With JavaScript, I have a much easier time planning and developing frameworks.
JavaScript runs on nearly every operating system, and an engine is included in mainstream web browsers.

ii) Python: I like it because it's a very popular programming language and because of its set of robust libraries that make it such a dynamic and a fast programming language. It's object-oriented, and it really allows for everything from creating a website, to app development, to creating different types of data models. Python is consistently ranked as one of the easiest programming language to learn and is known for its high reliability and simple syntax. With Python I can use it in a wide number of applications from AI, to video games, to productivity tools.

iii) Ruby: I understand Rails is long in the tooth and has deficiencies such as no concurrency but if I have to prototype a web app so quickly, I can hardly do better than Rails. It's still a popular framework in the surveys and has tons of features. If I want to move to Elixir & Phoenix from RoR, I understand but that is a new learning curve. That said, I too I am confused about why Ruby is being trashed like by many programmers. Ruby inspires loyalty like few languages and it works for many common problems that don't require concurrency or high performance. I would have a hard time beating Ruby's expressiveness within an object oriented context.

iv) PHP: I like PHP because of its popularity as a web development language; it integrates well with an existing stack of web tools (inline HTML, good Apache integration, built-in MySQL support, etc). I like it *NOT* because it's a well-designed language that is easy to work in; it's not, by the way.

v) Docker is likely high on my list of "liked" technologies because I see it show up more and more in job listings and I have concluded that I want to go work at a job where I can learn it, not necessarily that I already have used it and liked it.

vi) SQL: Its high speed advantage puts it to my tool box of favorite Programming languages. Using the SQL queries, I can quickly and efficiently retrieve large amount of records from a database. In the standard SQL, it is very easy to manage the database system. SQL doesn't require a substantial amount of code to manage the database system. SQL has well defined standards; long established standards are used by the SQL databases that are being used by ISO (International Organization for Standardization) and ANSI (American

National Standards Institute). Because of SQL portability, it can be used in laptop, PCs, server and even some mobile phones. SQL is an interactive language, therefore, SQL is a domain language used to communicate with the database. It is also used to receive answers to the complex questions in seconds. Using the SQL language, I can make different views of the database structure and all the above reasons make SQL a very likeable language to me.

vii) HTML (HyperText Markup Language): HTML is the language I use to tell a web browser what each part of a website is. So, using HTML, I can define headers, paragraphs, links, images, and more, so a browser knows how to structure the web page I am looking at. HTML can be used create only static and plain pages that can be viewed on the browser. So, I like HTML especially if I want to preview my work live on the browser though it has limitations to static and plain pages, so if I need dynamic pages then HTML is not useful.

viii) CSS (Cascading Style Sheet): CSS handles the look and feel part of a web page. Using CSS, I can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, etc.

The following are the main reasons why I like CSS in particular:

1) CSS saves time − I can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
2) Easy maintenance − To make a global change, I simply change the style, and all elements in all the web pages will be updated automatically.
3) Global web standards − Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.
4) Platform Independence − The Script offer consistent platform independence and can support latest browsers as well.

ix) Java: Java is a class based, object oriented programming that is designed to have as few implementation dependencies as possible. It is a general purpose programming language intended to let application developers write once and run everywhere, meaning that a compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture computer architecture. The syntax of Java is similar to C and C++, but has fewer low level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages, because of these privileges I really Java.

b) The following are a couple programming languages that I don't really dislike but are the areas I am yet to explore and why so:

i) Perl: Perl is a powerful programming language but of course, what I hear most is that it's hard to read and that's undeniable. I'd deny that, myself. It can be written so that it's harder to read, particularly if one doesn't make any effort to think about how it'll be read later.

ii) Visual Basic and Cobol are not a good career choice for me at the moment. I really have no much time to invest in this two.

iii) ASP.net: I find it very hard to get a good answer for most of my questions on ASP.net MVC.
The Visual studio IDE does not provide good support for UI development.
Most of the security mechanisms like Identity require long learning curve and often have easier implementation in other languages.
EDMX solves ORM problems but complicates things when DB schema changes in DB first approach. On multiple occasions, my project has gone bonkers when i used the update EDMX from updated DB option.
Error handling needs to be user defined and hard to figure out and most of the errors are directly showing in the View (Browser), with haphazard stack trace, which is difficult to understand.
When NuGEt packages are updated, several dlls are added which don't contribute to the code quality, they are mostly add-ons which don't serve any purpose unless some arcane function gets used in a corner case which is usually not required.

For these reasons and so much more, I completely dislike development in ASP.net.

iv) Clojure: It's usage is shrinking –It has a basic problem to my understanding, it doesn't look at all like C. No curly braces, No #MACROS, No int or char or *. In my own understanding, if a language doesn't look an awful a lot like C it's automatically not a good choice language. I never mind that it's better to other developers. I never mind that one can be more productive. I never mind that one can learn something just by figuring out how to use it. So far too many developers, anything that's new or different is bad. C is old and I spent a whole lot of time learning to do it (perhaps badly), so therefore C is good to me as well!